# Final Report for AFOSR #FA9550-08-1-0422
# *Landscape Analysis and Algorithm Development for Plateau Plagued Search Spaces*
# August 1, 2008 to November 30, 2010

Adele Howe L. Darrell Whitley
Computer Science Department
Colorado State University
Fort Collins, CO 80524
email: {howe,whitley}@cs.colostate.edu

February 28, 2011

**Abstract**

Over the last 10 to 20 years, heuristic search in the Operations Research and Artificial Intelligence communities has focused on developing high level general purpose algorithms, such as Tabu Search and Genetic Algorithms. However, understanding of when and why these algorithms perform well still lags. Our project extended the theory of certain combinatorial optimization problems to develop analytical characterizations of portions of search spaces and as the basis for creating new algorithms for two well known problems. We focused attention on two specific subclasses of NP-Hard problems: elementary landscapes, which include Traveling Salesman Problem (TSP) and embedded landscapes, which include Maximum Satisfiability (MAXSAT). Our analysis supports calculating exactly the statistical moments of the distributions of values in regions of MAXSAT search spaces and explains why true plateaus are rare. Our new algorithm for TSP is competitive with the state of the art, while being much simpler and easier to understand.

# Report Documentation Page

*Form Approved*
*OMB No. 0704-0188*



| 1. REPORT DATE | 2. REPORT TYPE | 3. DATES COVERED |
|---|---|---|
| **28 FEB 2011** | **Final** | **01-08-2008 to 30-11-2010** |

4. TITLE AND SUBTITLE
**LANDSCAPE ANALYSIS AND ALGORITHM DEVELOPMENT FOR PLATEAU PLAGUED SEARCH SPACES**

5a. CONTRACT NUMBER
**FA9550-08-1-0422**

5b. GRANT NUMBER

5c. PROGRAM ELEMENT NUMBER

6. AUTHOR(S)
**Adele Howe; L Whitley**

5d. PROJECT NUMBER

5e. TASK NUMBER

5f. WORK UNIT NUMBER

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)
**Colorado State University,Fort Collins ,CO,80523**

8. PERFORMING ORGANIZATION REPORT NUMBER
**; AFRL-OSR-VA-TR-2011-0225**

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)
**AFOSR, 875 North Randolph Street, Suite 325, Arlington, VA, 22203**

10. SPONSOR/MONITOR’S ACRONYM(S)

11. SPONSOR/MONITOR’S REPORT NUMBER(S)
**AFRL-OSR-VA-TR-2011-0225**

12. DISTRIBUTION/AVAILABILITY STATEMENT
**Approved for public release; distribution unlimited**

13. SUPPLEMENTARY NOTES

14. ABSTRACT
**Over the last 10 to 20 years, heuristic search in the Operations Research and Artificial Intelligence communities has focused on developing high level general purpose algorithms, such as Tabu Search and Genetic Algorithms. However, understanding of when and why these algorithms perform well still lags. Our project extended the theory of certain combinatorial optimization problems to develop analytical characterizations of portions of search spaces and as the basis for creating new algorithms for two well known problems. We focused attention on two specific subclasses of NP-Hard problems: elementary landscapes, which include Traveling Salesman Problem (TSP) and embedded landscapes, which include Maximum Satisfiability (MAXSAT). Our analysis supports calculating exactly the statistical moments of the distributions of values in regions of MAXSAT search spaces and explains why true plateaus are rare. Our new algorithm for TSP is competitive with the state of the art, while being much simpler and easier to understand.**

15. SUBJECT TERMS
**Operations Research, Artificial Intelligence, Search Algorithms**

| 16. SECURITY CLASSIFICATION OF: a. REPORT | b. ABSTRACT | c. THIS PAGE | 17. LIMITATION OF ABSTRACT | 18. NUMBER OF PAGES | 19a. NAME OF RESPONSIBLE PERSON |
|---|---|---|---|---|---|
| **unclassified** | **unclassified** | **unclassified** | **Same as Report (SAR)** | **22** | |

**Standard Form 298 (Rev. 8-98)**
Prescribed by ANSI Std Z39-18

# Contents

# 1 Project Objectives

The local search family of techniques have been surprisingly effective at finding competitive solutions for NP-hard combinatorial optimization problems. By and large, these techniques leverage local gradient information in the search space to incrementally improve solutions. However, the interaction between the search space topology and the meta-heuristic leads to some problem domains and instances being more difficult to solve than others.

Our long term research goal is to develop models that link local search performance to problem characteristics, especially those that are problematic for local search (local optima and plateaus). The goals in the proposal were to enhance the understanding of the relationship between local search and problem difficulty and to develop new algorithms that leverage that understanding. These goals led to three specific objectives for our project:

1. Extending the theory of local search, in particular, that of Elementary Landscapes,

2. Designing a new algorithm for a problem that possesses an elementary landscape: Traveling Salesman Problem (TSP),

3. Developing analytical techniques for characterizing problem instances that possess embedded landscapes, e.g., Maximum Satisfiability (MAXSAT).

We focused on Elementary Landscape theory because several NP classes of problems have been proven to have elementary landscapes, e.g., TSP, Min-Cut and Max-Cut Graph Partitioning, Graph Coloring, Frequency Assignment, Not-all-Equal-Sat and Weight Partitioning over N objects. Thus, new theoretical results for Elementary Landscape automatically translate to a suite of problems. Technically we should be clear, that search problems display an elementary landscapes under a *particular* local search neighborhood, just as local optima exist with respect to a particular local search neighborhood. Also, some NP-hard problems that do not directly have elementary landscapes can be expressed as a superposition of a small number of elementary landscapes.

We then more thoroughly examined two problems: TSP and MAXSAT. TSP possesses an elementary landscape, while MAXSAT possesses an embedded landscape. However, the embedded landscape of any MAX-kSAT problem can be expressed as a superposition of k elementary landscapes. Therefore, a MAX-3SAT problem can be expressed as a superposition of only 3 elementary landscapes, making it possible to leverage elementary landscape theory in useful ways. For TSP, we applied the theory to the development of new algorithmic approaches and developed techniques that are much simpler and easier to understand than prior approaches, while also offering a high level of performance. For MAXSAT, we leveraged the embedded landscape and elementary landscape theory to construct proofs of certain properties (e.g., plateau width) and develop analytical techniques for efficiently calculating statistical properties of regions of search spaces within problem instances.

# 2 Accomplishments/New Findings

We extended the theory of elementary landscapes, applied the theory to deriving properties of well-known NP-hard problems, e.g., TSP, graph coloring, frequency assignment and k-satisfiability (k-SAT) and explored how knowledge of those properties can improve search. As indicated in the last section, we had three objectives to our research. This section is organized by those objectives.

### 2.1 An Introduction to Elementary Landscapes

We formally define elementary landscapes as follows. Let $X$ be a set of solutions, $f : X \to \mathbb{R}$ be a fitness function, and $N : X \to \mathcal{P}(x)$ be a neighborhood operator. Thus, $N(x)$ defines the set of points that are neighbors of solution $x$. The neighborhood operator can also be represented by its *adjacency matrix*:

$$\mathbf{A}_{xy} = \begin{cases} 1 & \text{if } y \in N(x) \\ 0 & \text{otherwise} \end{cases}$$

Since a discrete function over the set of candidate solutions $g : X \to \mathbb{R}$ can be characterized as a vector in $\mathbb{R}^{|X|}$, any $|X| \times |X|$ matrix can be used as a linear operator on that function. We will restrict our attention to *regular* neighborhoods, where $|N(x)| = d$, a fixed constant for all $x \in X$. When a neighborhood is regular, the *Laplacian operator* is $\Delta = \mathbf{A} - d\mathbf{I}$ and its influence on the fitness function $f$ is:

$$\Delta f(x) = \sum_{y \in N(x)} (f(y) - f(x))$$

Stadler defines the class of *elementary landscapes* where the objective function $f$ is an eigenfunction of the Laplacian of the graph induced by the neighborhood operator [13]. If function $f$ has been centered to have zero mean, then $-k$ is the eigenvalue of $\Delta f$.

$$\Delta f = k(\bar{f} - f) = -k(f)$$

Note that in general $f$ need not have zero mean, and $\Delta f = k(\bar{f} - f)$. For all elementary landscapes it is possible to compute $\bar{f}$, the average evaluation over the entire search space. We can also compute $\text{Avg}(N(x))$, the average evaluation over all of the neighbors of any solution $x$.

$$\begin{aligned} \text{Avg}(N(x)) &= \frac{1}{d} \sum_{y \in N(x)} f(y) = \frac{1}{d} \left( \sum_{y \in N(x)} f(y) - f(x) \right) + f(x) = \frac{1}{d} \Delta f(x) + f(x) \\ &= f(x) + \frac{k}{d} (\bar{f} - f(x)) \end{aligned}$$

From this equation one can derive many fundamental properties of elementary landscapes by simple algebra. One common property of elementary landscapes is that either

$$f(x) < \text{Avg}(N(x)) < \bar{f} \quad \text{or} \quad f(x) > \text{Avg}(N(x)) > \bar{f}$$

as long as $f(x) \neq \text{Avg}(N(x))$ and $0 < k/d < 1$. Furthermore, $f(x) \neq \text{Avg}(N(x))$ unless $f(x) = \text{Avg}(N(x)) = \bar{f}$. This means that all neighborhoods have either improving or disimproving moves, unless the *entire* search space is flat [2]. We have been able to leverage this result to prove that MAXSAT neighborhoods can never be completely flat unless $f(x)$ is very close to $\bar{f}$ [14].

#### 2.1.1 Our Component Based Model of Elementary Landscapes

For the elementary landscapes we have examined (TSP, Graph Coloring, Min-Cut and Max-Cut Graph Partitioning, Basic Frequency Assignment and select pseudo-Boolean functions), the "components" that make up a solution $x$ can be decomposed [3, 18, 21, 20]. Typically, these components are weights in a cost matrix (e.g., distance matrix in TSP). The components of the neighborhood $N(x)$ can be separated into two parts: 1) those weights that contribute to $f(x)$ and 2) those weights that do not contribute to $f(x)$. We exploit this property in identifying elementary landscapes and in efficient computation of the search space metrics.

Let $C$ denote the set of components used to construct the cost function. We can also view solution $x$ as just a subset of cost components. By a slight abuse of notation, we will let $(C-x)$ refer to the subset of components in $C$ that do not contribute to $f(x)$. When transforming $x$ to some $y \in N(x)$, we subtract a subset of components from $x$ and add new components from $(C-x)$ to the remaining components of $x$ to compute costs of $y$. If a landscape is elementary, all components in $x$ are uniformly sampled for potential removal from $x$ and all components in $(C-x)$ are uniformly sampled in the set of neighbors denoted by $N(x)$.

The following three ratios, $p_1$, $p_2$ and $p_3$, correspond to changes in sampling rates. Let $p_1$ denote the proportion of components in $f(x)$ that change when a move is made; $p_2$ is the proportion of components in $(C-x)$ that change when a move is made. And $p_3$ is the proportion of the total components in $C$ that contribute to the cost function for any randomly chosen solution such that $\bar{f} = p_3 \sum_{c \in C} c$. For $p_1$, $p_2$ and $p_3$ to be valid, sampling of the cost components must be *uniform*. For example, every weight in the cost matrix appears a uniform (equal) number of times across the space of all possible solutions. Every weight in $f(x)$ appears a uniform number of times in every neighborhood and every weight in $(C-x)$ appears a uniform number of times in every neighborhood. This is the critical insight that explains *why* we can use the eigenvalue from the Laplacian to compute neighborhood averages. The sum of the cost components in the set $(C-x)$ can be calculated by $(\sum_{c \in C} c) - f(x)$.

$$\mathrm{Avg}_{y \in N(x)}(f(y)) = f(x) - p_1 f(x) + p_2((\sum_{c \in C} c) - f(x)) \tag{1}$$

Assume the neighborhood size is $d$. Both $p_1$ and $p_2$ can be expressed relative to $d$; $p_3$ is independent of the neighborhood size.

**Theorem 1.** *Assume $p_1$, $p_2$ and $p_3$ can be defined over any regular landscape such that the evaluation function can be decomposed into components where*

$$p_1 = \alpha/d \quad \textit{and} \quad p_2 = \beta/d \quad \textit{and} \quad \bar{f} = p_3 \sum_{c \in C} c = \frac{\beta}{\alpha + \beta} \sum_{c \in C} c$$

*and $p_1, p_2$ and $p_3$ correspond to uniform sampling rates of the components; then the landscape is elementary and*

$$\mathrm{Avg}_{y \in N(x)}(f(y)) = f(x) + \frac{\alpha + \beta}{d}(\bar{f} - f(x)) \qquad \textit{where } k = \alpha + \beta$$

Our proof follows from simple substitution into equation 1 [20]. An example will illuminate these ideas. Consider the neighborhood for a 5 city TSP using the standard 2-opt neighborhood (see Table 1). The set $C$ contains all of the costs (edges) in the cost (distance) matrix. Let $w_{ab}$ denote an edge between vertices A and B. One can see from this example that the sampling rates are uniform over edges in $x$ and $C-x$.
Therefore, for the TSP under 2-opt we observe:

$$p_1 = \frac{2}{n} = \frac{n-3}{n(n-3)/2} \quad \text{and} \quad p_2 = \frac{2}{n(n-3)/2}$$

| | Edges in $x$ | | | | | Edges in $(C-x)$ | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| | $w_{ab}$ | $w_{bc}$ | $w_{cd}$ | $w_{de}$ | $w_{ae}$ | $w_{ac}$ | $w_{ad}$ | $w_{bd}$ | $w_{be}$ | $w_{ce}$ |
| $x$ = ABCDE | 1 | 1 | 1 | 1 | 1 | 0 | 0 | 0 | 0 | 0 |
| $y_1$ = ABEDC | 1 | 0 | 1 | 1 | 0 | 1 | 0 | 0 | 1 | 0 |
| $y_2$ = ABCED | 1 | 1 | 0 | 1 | 0 | 0 | 1 | 0 | 0 | 1 |
| $y_3$ = ABDCE | 1 | 0 | 1 | 0 | 1 | 0 | 0 | 1 | 0 | 1 |
| $y_4$ = ACBDE | 0 | 1 | 0 | 1 | 1 | 1 | 0 | 1 | 0 | 0 |
| $y_5$ = ADCBE | 0 | 1 | 1 | 0 | 1 | 0 | 1 | 0 | 1 | 0 |
| CHANGE | -2 | -2 | -2 | -2 | -2 | 2 | 2 | 2 | 2 | 2 |

Table 1: How edges uniformly move in and out of solution $x$ and the neighborhood $N(x)$.

because 2 edges are changed by 2-opt, and there are $n$ edges in solution $x$ and there are $n(n-1)/2 - n = n(n-3)/2$ edges in the set $C - x$. We next calculate $k$ and the neighborhood average

$$k = \alpha + \beta = n - 3 + 2 = n - 1$$

$$\text{Avg}_{y \in N(x)}(f(y)) = f(x) + \frac{n-1}{n(n-3)/2}(\bar{f} - f(x))$$

This result easily generalizes to all K-opt move operators and neighborhoods for the TSP.

### 2.1.2 Partial Neighborhoods of Elementary Landscapes

The key insight behind the concept of partial neighborhoods is that when $\alpha > \beta$ the components in $x$ are changing at a higher frequency than the components in $(C - x)$. When this is the case, the neighborhood can be subdivided; how the neighborhood is subdivided is not important, as long as the ratio $p_1$ still holds and all of the components that make up solution $x$ are uniformly sampled (included or removed) in each partial neighborhood. Furthermore, the ratio $\alpha/\beta$ measures how many partial neighbors exist if the partial neighborhoods are uniform in size.

In the TSP for example, when the number of cities denoted by $n$ is odd, $\alpha/\beta = \lfloor n/2 - 1 \rfloor$ yields the exact number of partial neighborhoods where $p_1$ holds. When $n$ is even, there is one degenerate partial neighborhood.

While these partial neighborhoods uniformly sample the components in $x$, they sample only a subset of the components of $C - x$. The question then becomes whether the sampling of $C - x$ is regular enough to be concisely described. So in this case, how the neighborhood is subdivided becomes important.

The key question is this: how can we partition the 2-opt neighborhood so that in a given partition every component of $x$ is uniformly represented? It is easier to answer a complimentary form of this question: how can we select a subset of 2-opt moves that uniformly removes edges from $x$? If edges are uniformly removed, they will also be uniformly represented in the partial neighborhood.

The answer is to group all of the 2-opt moves according to the length of the segment that is reversed. For example, we can group together all of the 2-opt moves that reverse a segment of length 3. This works because we can reverse a segment starting at city 1, then at city 2, then at city 3 and so on until we reverse a segment at city $n$. Therefore every edge in solution $x$ is cut exactly twice: when we cut at location $i$ we also cut at location $i + 3$ and our index wraps around the TSP tour (Hamilton circuit).

Let M correspond to a new upper triangle matrix where the indices of $M$ also index a permutation $x$ representing a tour in a TSP. When interpreted as edges, $M$ corresponds to all of the edges in the cost matrix. Each edge in $M$ where $j = i+1$ corresponds to an edge found in $x$. Each edge in $M$ where $j \neq i+1$ corresponds to an edge found in the 2-opt neighborhood of $x$ that is not found in $x$, since if they are not in $x$ they are edges in the set (C - x). If $j-i = l$ or $j-i = n-l$ then the 2-opt move that generated edge $e_{i,j}$ was produced by reversing a segment of length $l$. This means the diagonals where $j-i = l$ or $j-i = n-l$ corresponds to a partial neighborhood that uniformly samples edges in solution $x$. What we will actually "store" in our matrix is the partial neighborhood to which edge $m_{i,j}$ belongs. We next consider a 13 city example.

| | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 1 | | X | A | B | C | D | E | E | D | C | B | A | X |
| 2 | | | X | A | B | C | D | E | E | D | C | B | A |
| 3 | | | | X | A | B | C | D | E | E | D | C | B |
| 4 | | | | | X | A | B | C | D | E | E | D | C |
| 5 | | | | | | X | A | B | C | D | E | E | D |
| 6 | | | | | | | X | A | B | C | D | E | E |
| 7 | | | | | | | | X | A | B | C | D | E |
| 8 | | | | | | | | | X | A | B | C | E |
| 9 | | | | | | | | | | X | A | B | C |
| 10 | | | | | | | | | | | X | A | B |
| 11 | | | | | | | | | | | | X | A |
| 12 | | | | | | | | | | | | | X |

Now we have segments of length 2, 3, 4, 5 and 6. And since $n$ is odd, there are $\alpha/\beta = 13 - 3/2 = 5$ which correspond to the 5 different lengths, and each partial neighborhood is made up of $n$ distinct edges. The edges marked by `X` correspond to the current solution. Edges marked by `A` belong to a partial neighborhood corresponding to a 2-opt move of length 2. In general, the $i^{th}$ character of the alphabet denotes a partial neighborhood made up of 2-opt moves of length $i+1$. We have developed a relatively simple indexing scheme that allows us to compute updates to statistics over these partial neighbors on demand at reasonable cost as the search moves from one incumbent solution to the next.

Thus, we can ask and answer the question, which partial neighborhood has the highest expectation of yielding an improving move?

### 2.1.3 Graph Coloring and Partial Neighborhoods

We first use the component model to show that graph vertex coloring is elementary under the vertex recoloring operator. We have also shown that Weighted Vertex Coloring and the basic Frequency Assignment Problem are elementary [18].

Let $G$ be a graph, $V$ the set of vertices, and $E$ the set of edges. The graph coloring problem assigns one of $r$ number of colors to the vertices of a graph. A conflict exists if two vertices connected by an edge have the same color. The evaluation function $f(x)$ counts how many connected vertices have the same color. The set of components $C$ corresponds to the set of edges. Every edge either contributes cost 1 to the cost function if the vertices connected by that edge have the same color, or the edge contributes 0 to the cost function if the vertices connected by the edge have a different color. In addition to the edges that contributed to $f(x)$, there are $|E| - f(x)$ edges in $C$ that do not contribute to $f(x)$.

The neighborhood operator is to recolor every vertex in the graph. Since there are $|V|$ vertices, and each vertex can be recolored in $r-1$ ways, the size of the neighborhood is $d = |V|(r-1)$. The average cost over all solutions will be

$$\bar{f} = 1/r \sum_{c \in C} c = |E|/r \quad \text{where: } 1/r = p_3$$

Consider two vertices $v_1$ and $v_2$ that are the same color and connected by an edge. There are $r-1$ colors that can be assigned to either $v_1$ or $v_2$ that will remove the conflict. Therefore:

$$p_1 = 2(r-1)/\left(|V|(r-1)\right) = \frac{\alpha}{d}$$

When a conflict does not exist, there are exactly two ways for the conflict to be generated: either $v_1$ is colored the same as $v_2$, or $v_2$ is colored the same as $v_1$. Therefore:

$$p_2 = 2/\left(|V|(r-1)\right) = \frac{\beta}{d}$$

which yields

$$\text{Avg}(N(x)) = f(x) + \frac{2r}{|V|(r-1)}(\bar{f} - f(x)) \quad \text{where } k = 2r.$$

When attempting to remove conflicts in a graph coloring problem, it makes no sense to consider every vertex in the graph. Assume vertices $v_i$ and $v_j$ are not in conflict with any other vertices in the graph. Then changing the colors of $v_i$ and $v_j$ is useless. Recoloring these vertices cannot remove existing conflicts, but may generate new conflicts. But if they are not included in the neighborhood, the elementary landscape collapses.

Any smart local search algorithm will restrict moves to those that change vertices that are involved in a conflict. This induces a partial neighborhood. In most cases the expected value of this partial neighborhood will be improved compared to the full neighborhood. (It is unintuitive, but one can construct examples where this "smart" strategy is inferior to exploring the full neighborhood.) We can define a partial neighborhood as follows. Let $Degree(v)$ be a vector that stores the degree of every vertex $v \in |V|$. Let $Q_x$ be the set of vertices such that if edge $e_{i,j}$ contributes cost to $f(x)$ then vertices $i$ and $j$ are members of set $Q_x$.

**Theorem 2.** *The partial neighborhood for the graph coloring problem, denoted by $N'(x)$, such that only vertices in $Q_x$ are recolored, is given by:*

$$\text{Avg}_{y \in N'(x)}(f(y)) = f(x) + \frac{\left[\left(\sum_{v \in Q_x} Degree(v)\right) - (2r)f(x)\right]}{|Q_x|(r-1)}$$

The proof [20] directly exploits the component model. (The theorem and proof can be generalized to the weighted graph coloring problem by constructing a corresponding *Weight* vector in place of the simple *Degree* vector [18].) This partial neighborhood is not elementary because it is dynamically defined with respect to $x$. Nevertheless the average value of all of the neighbors in the dynamically defined neighborhood can be cheaply computed by exploiting a decomposition of the full vertex coloring elementary landscape neighborhood. We can also prove

$$\sum_{v \in Q_x} Degree(v) < (r+2)f(x) \quad \Longleftrightarrow \quad \underset{y \in N'(x)}{\text{Avg}}(f(y)) < f(x).$$

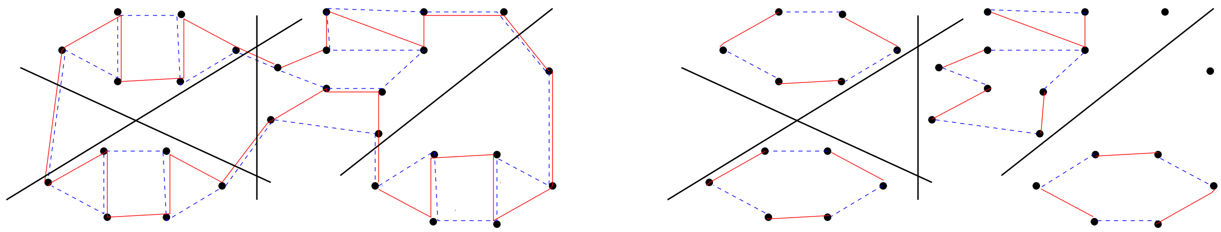

Figure 1: An example of a graph $G$ created from the union of two parent tours. By deleting common edges, we break this graph into 4 independent subgraphs.

When this happens an improving move is guaranteed to exist. This kind of information might be used to create better forms of local search that automatically detect that certain improving moves are available, or that can calculate and compare the probability of the existence of an improving move in different neighborhoods.

We have also shown that MAX-CUT and MIN-CUT graph partitioning problems are elementary landscapes and have partial neighborhoods that focus attention on potential improving moves, while eliminating non-improving moves [20].

### 2.2 Elementary Landscapes as Constrained Linear Systems

A key characteristic of elementary landscapes is that the objective function is linear, but with feasibility constraints. But is there some way to exploit this linearity? In a sense, local search methods have been exploiting this linearity for decades; but can we do better?

We will take two solutions and ask how they can be decomposed into linear independent subsolutions without violating the feasibility constraints. This process will look a great deal like a "recombination" from the genetic algorithms literature; and in fact we will describe our operator as a "crossover" operator. We will exploit a localized decomposition of local optima for the TSP, and show how it is possible to tunnel directly from local optimum to local optimum.

#### 2.2.1 Generalized Partition Crossover

We construct a graph $G = (V, E)$ where $V$ is the set of vertices (i.e., cities) of an instance of a TSP and $E$ is the union of the edges found in *two* different candidate solutions. An edge in $E$ is a *common* edge if it is found in both parents; an edge is an *uncommon* edge if it is found in only one parent. Next, we cut G into two graphs, by cutting only 2 edges (such a cut is said to have cost 2). One can easily prove that a cut of cost 2 must only cut common edges, otherwise the cost would be greater than 2; thus such a cut divides G into independent subproblems.

*Generalized partition crossover* (GPX) exploits all partitions of cost 2 in a single recombination in $O(N)$ time [19]. We recombine solutions by creating a subgraph of $G$, $G_u = (V, E_u)$, where $V$ is the vertex set of the original TSP instance and $E_u$ is the set of uncommon edges found in $E$. Ideally, $G_u$ comprises multiple disconnected and independent subgraphs. The lefthand side of figure 1 shows a graph $G$ created from two parents. The edges from one parent are represented by solid lines and those from the other parent by dashed lines. On the righthand side of figure 1 is the same graph with the common edges deleted (i.e., graph $G_u$); this breaks the graph into 4 subgraphs. Multiple partitions of this graph have cost 2 (shown by the heavy dark lines).

We use Breadth First Search on $G_u$ to find each connected subgraph of $G_u$; this has $O(N)$ cost, because the degree of any vertex is at most 4, and each vertex is processed only once. Finding all the cuts of cost 2 breaks the graph $G$ into $k$ pieces which we call *partition components*; not all connected components in $G_u$ yield feasible partition components because they may not yield cuts of cost 2. We then prove the following result [19]:

**The GPX Theorem:**

*Let graph $G$ be constructed by unioning the vertices and edges found in two Hamiltonian Circuits for some instance of the TSP. If graph $G$ can be separated into $k$ partition components using only cuts of cost 2, then there are $2^k - 2$ possible distinct offspring. Every potential offspring inherits all the common edges found in the parents, and is composed entirely of edges found in the two parents. If the parents are locally optimal, then every partition component that is inherited is "piecewise" locally optimal.*

Because the subpath solutions manipulated by GPX are linearly independent, GPX can be applied in a greedy fashion, selecting the best subsolution from each partition component. Therefore, the power of GPX is that it can "filter" large numbers of local optima in a single $O(N)$ recombination step.

GPX is not guaranteed to be feasible, but GPX is feasible with extremely high frequency when the solutions being recombined are local optima. We have tested GPX on random local optima generated using 2-opt [5], 3-opt and a variant of Lin-Kernighan search [11] (LK-search) on various problems from the TSPLIB (see [19] for details); GPX was feasible in 100% of the cases when combining these local optima. Moreover, if the parents are locally optimal under any move operator, the subpaths that are inherited from the parents are piecewise locally optimal under the same operator (whatever the operator!). Thus, the only way an "offspring" can be improved is by a move that exchanges edges from *different* partition components. As a result, the majority of offspring produced by GPX (typically more than 90% of the time) are also locally optima.

To show that GPX can filter thousands and even millions of local optima, we applied 3-opt and LK-search on problems ranging in size from 532 to 1817 cities. For 3-opt, the average number of partitions ranged from k=10 (for ATT532) to k=26 (for u1817). At k=26, one GPX recombination was filtering more than 67 million solutions, the majority of which were local optima. For LK-search, the number of partitions ranged from k=5 to k=13. GPX also displays excellent scaling: the larger the problem, the larger the number of partitions that are found.

#### 2.2.2 GPX Experimental Results

We embedded GPX in a very simple genetic algorithm (GA) using a population of only 10. Every solution is improved (when possible) using 1 pass of LK-Search as implemented in the Concord package [1]. We then compared the results to Chained Lin-Kernighan (Chained-LK), which also uses exactly the same LK-Search with identical parameter settings. Chained-LK is one of the better performing local search heuristics for the TSP [1]. Chained-LK applies LK-search to a single tour, uses a double bridge move [9] to perturb the solution and then reapplies LK-search. Since the population size is 10, the GA+GPX uses 10 applications of LK-search each generation; therefore, Chained LK is allowed to do 10 double-bridge moves and 10 LK-search improvements for every generation executed by the GA+GPX. Both algorithms call LK-search exactly the same number of times. Table 2 lists the average percentage of the cost of the minimum tour found compared to the cost of the global optimum for each problem

| Generation ⟶ | | 10 | 20 | 50 | 100 |
|---|---|---|---|---|---|
| Instance | Algorithm | 110 LK calls | 210 LK calls | 510 LK calls | SOLVED |
| att532 | GA+GPX | $0.18 \pm 0$ | $0.12 \pm 0$ | $0.07 \pm 0$ | 26/50 |
| | Chained-LK | $0.21 \pm 0.01$ | $0.13 \pm 0$ | $0.08 \pm 0$ | 16/50 |
| nrw1379 | GA+GPX | $0.48 \pm 0$ | $0.34 \pm 0$ | $0.23 \pm 0$ | 1/50 |
| | Chained-LK | $0.46 \pm 0.01$ | $0.32 \pm 0$ | $0.19 \pm 0$ | 1/50 |
| rand1500 | GA+GPX | $0.52 \pm 0.01$ | $0.36 \pm 0$ | $0.22 \pm 0$ | 12/50 |
| | Chained-LK | $0.54 \pm 0.01$ | $0.39 \pm 0.01$ | $0.25 \pm 0$ | 2/50 |
| u1817 | GA+GPX | $1.26 \pm 0.01$ | $0.95 \pm 0.01$ | $0.63 \pm 0.01$ | 1/50 |
| | Chained-LK | $1.61 \pm 0.02$ | $1.19 \pm 0.01$ | $0.83 \pm 0.01$ | 0/50 |

Table 2: Columns marked 10 to 50 show the average percentage of the cost of the minimum tour found above the globally optimal cost averaged over 500 experiments using Chained LK and GA+GPX. SOLVED shows how often each algorithm found an optimal solution.

| Instance | Metric | LK-Helsgaun | GA+GPX |
|---|---|---|---|
| u1817 | Percent Over Optimal: | 1.0005 ± 0.00033 | 1.00007 ±0.0001 |
| u1817 | Optimal found: | 13/100 | 59/100 |
| fea5557 | Percent Over Best Known: | 1.00004 ± 0.00008 | 1.00009 ± 0.00006 |
| fea5557 | Best Known Found: | 58/100 | 16/100 |

Table 3: Results after 100 trials of 10,000 LK search calls.

instance. The GA+GPX was allowed to run for 100 generations in these experiments.

GA+GPX yields better results on all of the problems except nrw1379. This is remarkable because the Hybrid GA must optimize 10 solutions and the best solution must be optimized to be 10 times faster than Chained-LK to obtain a better result with the same effort.

If each algorithm is run longer, the performance of GA+GPX is increasingly better than Chained LK. The last column of Table 2 (SOLVED) shows how many times out of 50 attempts that each method finds the global optimum after 1010 calls to LK-search.

### 2.2.3 What about LK-Helsgaun on larger TSP Instances?

The LK-Helsgaun algorithm [7, 8](LKH) appears to be the most likely candidate for best TSP solver in the world. So how do we compare? Our data suggest GA+GPX is competitive. The LKH algorithm uses a more expensive form of LK-search (compared to Chained-LK) that allows additional 4-opt and 5-opt moves. For GA+GPX, we again use a population of only 10 solutions that were each improved using the same LK-search code used by LKH; after each application of GPX, we also used the LK-search code of LKH to try to improve the output from GPX. We ran for 10,000 LK search calls to try to find optimal (for u1817) or best known (for fea5557) solutions. In terms of optimal solutions found, GA+GPX is superior to LKH on u1817, but on fea5557, the results are reversed (see Table 3). The percents over best known are very close.

While there is no clear winner, GPX still has two advantages over LKH. First, GPX is very simple, and we know why it works. LKH uses several complex mechanisms to escape local optima. It is not really clear why these mechanisms work other than the fact that they randomize the search *just enough*. But it can still take a while to "get lucky" when randomizing a search.

Second, GA+GPX is remarkably effective at identifying the edges in the globally optimal solution. On every problem we have looked at with less than 2000 cities, all the edges needed to reconstruct the global optimum are found in the population with very high reliability after only 50 calls to LK-Search.

We are just starting to look at very large problems. Our preliminary data for fea5557 shows that on average we find 99.93% of the edges in the global optimum after only 100 recombinations. We have also determined empirically that we can get 100% of the edges on the majority of runs if we "accumulate edges". This means that we look at the best solution over a small number of time steps. For example if we keep the 2 previous best found solutions in addition to the population of size 10, we find 100 percent of the edges in the global optimum the majority of the time.

Of course one could also run Chained-LK or LKH multiple times and then try to find the solution using dynamic programming. This has already been done by Cook and Seymour, and it was very successful [4]. But they required considerably more effort. We explored this by running Chained-LK and LKH 10 times, for a total of 1,000,000 LK-search calls on fea5557, which is 1000 times more effort than needed by GA+GPX. Yet, the results are decidedly poorer; on average, LKH only finds 99.6% of the edges found in the best known solution, compared to 99.93% for GA+GPX.

We see many opportunities for the ideas behind GPX to be used in other domains. For example, our methods could be used to build better vehicle routing algorithms, which would impact a wide range of industries and military applications.

### 2.3 MAX-kSAT as a Composition of Elementary Landscapes

Rana and Whitley [12] show how to use a discrete Fourier Transform (in the form of a Walsh Transform, or Hadamar Transform) to express the MAXSAT evaluation function in polynomial form. We can use this result to obtain a decomposition of the MAX3SAT objective function into elementary components. The neighborhood $N(x)$ is the standard Hamming (bit-flip) neighborhood.

We will begin with a simple example. Assume we have a simple MAX-3SAT problem with 2 clauses and 4 variables.

$$f(x) = (\neg x_2 \vee x_1 \vee x_0) + (x_3 \vee \neg x_1 \vee \neg x_0)$$

Let $f_c$ denote the evaluation function for one of the clauses and let $f_x$ denote the evaluation function for the other clause such that

$$f_c(x) = (\neg x_2 \vee x_1 \vee x_0) \qquad \text{and} \qquad f_z(x) = (x_3 \vee \neg x_1 \vee \neg x_0)$$

We can express the evaluation function $f_c$ as a vector and assume $x$ is a bit vector of length 3 to be evaluated (we will ignore the fact that $x$ is a substring): $f_c =< 1, 1, 1, 1, 0, 1, 1, 1 >$. We then use the Walsh matrix denoted by $W$ to extract the Walsh coefficients from $f_c$.

$$\frac{1}{2^L} f_c W =< w_0, w_1, w_2, w_3, w_4, w_5, w_6, w_7 >$$

Given two bit strings $x$ and $y$ of length $n$, we denote the inner product $\langle x, y\rangle$ as $\sum_{b=1}^{n} x[b]y[b]$. We define the $i^{\text{th}}$ Walsh function $i = \{0, \ldots, 2^n - 1\}$ as

$$\psi_i(x) = (-1)^{\langle i,x\rangle}$$

Here, the $i$ that appears in the inner product of the exponent is taken to be the *bit string representation* of the index $i$, that is, the binary sequence of length $n$ that corresponds to the integer $i$. Next we use the Walsh functions and coefficients to represent $f_c$ as a polynomial.

$$\begin{aligned} f_c(x) &= w_0 + w_1\psi_1(x) + w_2\psi_2(x) + w_4\psi_4(x) + \\ &\quad w_3\psi_3(x) + w_5\psi_5(x) + w_6\psi_6(x) + w_7\psi_7(x) \end{aligned}$$

We can also decompose $f_c$ into 3 functions:

$$\begin{aligned} \text{Let } f_c(x) &= f_{c1}(x) + f_{c2}(x) + f_{c3}(x) \\ f_{c1}(x) &= w_0 + w_1\psi_1(x) + w_2\psi_2(x) + w_4\psi_4(x) \\ f_{c2}(x) &= w_3\psi_3(x) + w_5\psi_5(x) + w_6\psi_6(x) \\ f_{c3}(x) &= w_7\psi_7(x) \end{aligned}$$

It is easy to prove that $f_{c1}$ and $f_{c2}$ and $f_{c3}$ are all elementary. The functions $f_{c1}$ and $f_{c2}$ and $f_{c3}$ represent the linear terms, the pairwise terms, and the third order terms, respectively. We can do the same for the clause $f_z$,

$$f_z(x) = f_{z1}(x) + f_{z2}(x) + f_{z3}(x)$$

where $f_{z1}$ and $f_{z2}$ and $f_{z3}$ are also elementary landscapes.

We next combine this to compute a polynomial for the MAX-3SAT function $f$, where

$$f = f_c + f_z = (f_{c1} + f_{z1}) + (f_{c2} + f_{z2}) + (f_{c3} + f_{z3})$$

The combined subfunction $(f_{c1} + f_{z1})$ is still an elementary landscape because $f_{c1}$ and $f_{z1}$ have the same neighborhood size and eigenvalue. Similarly, both $(f_{c2} + f_{z2})$ and $(f_{c3} + f_{z3})$ are also elementary landscapes.

We can generalize this result to show that every MAX-3SAT problem is a superposition of at most 3 elementary landscapes. We can further generalize this result to show that for any MAX-kSAT problem:

$$\text{Avg}_{y \in N(x)}(f(y)) = \sum_{p=0}^{k} \left(1 - \frac{2p}{n}\right) \sum_{i:\langle i,i \rangle = p} w_i \psi_i(x)$$

We generalize our results using Walsh polynomials. Every MAX-3SAT objective function $f$ can be written as

$$f(x) = \sum_i w_i \psi_i(x) \quad \text{where} \quad w_i = \sum_{j=1}^{m} w_{i,c_j}$$

and where $w_{i,c_j}$ is the contribution to $w_i$ from clause $c_j$. We can again use the Walsh matrix denoted by $W$ to extract the Walsh coefficients from the evaluation subfunction from clause $c_j$, but we have also derived closed form equations that directly compute the Walsh coefficients [6, 12].

We next address the status of MAX-kSAT as an elementary landscape. We define a $|X| \times |X|$ *Markov transition matrix* $\mathbf{T}$

$$\mathbf{T}_{xy} = \begin{cases} \frac{1}{|N(x)|} & \text{if } y \in N(x) \\ 0 & \text{otherwise} \end{cases}$$

This matrix quantifies the transition probabilities between states on a random walk of the graph of the state space induced by the neighborhood operator $N(x)$.

One can prove that the Walsh function $\psi_i$ of order $\langle i, i\rangle = p$ is an eigenvector of the Markov transition matrix $\mathbf{T}$ with eigenvalue $\left(1 - \frac{2p}{n}\right)$. We define $\varphi^{(p)}$ as the *Walsh span* of order $p$ where $\varphi^{(p)}(x) = \sum_{i:\langle i,i\rangle=p} w_i\psi_i(x)$. One can then show that the $p^{\text{th}}$ Walsh span is an elementary landscape.

Since $\varphi^{(p)}$ is an eigenfunction of $\mathbf{T}$:

$$\mathbf{T}\varphi^{(p)} = \mathbf{T}\left[\sum_{i:\langle i,i\rangle=p} w_i\psi_i\right] = \sum_{i:\langle i,i\rangle=p} w_i\left(1-\frac{2p}{n}\right)\psi_i = \left(1-\frac{2p}{n}\right)\left[\sum_{i:\langle i,i\rangle=p} w_i\psi_i\right] = \left(1-\frac{2p}{n}\right)\varphi^{(p)}$$

On any MAX-kSAT instance, the expectation of the random variable $Y$ is then a linear combination of the $k+1$ Walsh spans evaluated at $x$.

$$\text{Avg}_{y\in N(x)}(f(y)) = \sum_{p=0}^{k}\left(1-\frac{2p}{n}\right)\varphi^{(p)}(x) = \sum_{p=0}^{k}\left(1-\frac{2p}{n}\right)\sum_{i:\langle i,i\rangle=p} w_i\psi_i(x)$$

Since $\psi_i$ is constant, this is the composition of $k$ elementary landscapes; hence, MAX-3SAT is a superposition of 3 elementary landscapes.

### 2.3.1 Generalized Neighborhoods and Generalized Statistical Moments

In general we can compute the $c^{th}$ moment for the fitness distribution using products of the Walsh coefficients

$$\begin{aligned}\mu_c &= \frac{1}{2^L}\sum_{a_1\oplus a_2\oplus\ldots\oplus a_c=0} w_{a_1}w_{a_2}\ldots w_{a_c}2^L, \qquad a_i \neq 0\ \forall\ i \\ &= \sum_{a_1\oplus a_2\oplus\ldots\oplus a_c=0} w_{a_1}w_{a_2}\ldots w_{a_c}, \qquad a_i \neq 0\ \forall\ i \qquad (2)\end{aligned}$$

This formula allows us to compute the variance, skew and kurtosis for any fitness distribution provided we are given the Walsh coefficients.

$$variance = \mu_2 = \sigma^2 \quad skew = \frac{\mu_3}{\sigma^3} \quad kurtosis = \frac{\mu_4}{\sigma^4}$$

For example, since $a_1 \oplus a_2 = 0$ if and only if $a_1 = a_2$ then the variance for any function can be computed

$$\sum_{i=1}^{2^L-1} w_i w_i$$

Of course, this computation of the moment around the mean, if done directly, would take exponential time. However, for MAXSAT only a polynomial number of Walsh coefficients are nonzero and only the nonzero coefficients need be considered. Selecting the indices to have even parity would consist of selecting the first $c-1$ indices from the set of nonzero Walsh coefficients. The exclusive-or of these would be taken and would be used as the desired $c^{th}$ index. The exclusive-or of the $c$ indices would therefore be zero. Using this simple strategy it would take $O(m^{c-1})$ time to compute the $c^{th}$ moment given the Walsh coefficients, where $m$ is the number of clauses in the MAXSAT problem. (There are also $O(m)$ nonzero Walsh coefficients.)

#### 2.3.2 Beyond the Immediate Neighborhoods

We can further extend our computation from neighborhoods to localized Hamming spheres: neighborhoods within some radius of the current state. Again, only the nonzero Walsh coefficients need be considered and there are at most $8m$ of these for a MAX-3SAT problem with $m$ clauses.

Given a Hamming sphere at distance $r$ (the radius) from point $x$, all of its neighbors are either at distance $r-1$ or at distance $r+1$. The information needed to figure out what points are in which set of neighbors is found in the adjacency matrix $A$. A point at distance $r$ has exactly $r$ bits that differ from $x$. We can calculate the average at distance $r$ in terms of matrix $A$, then reformulate the calculation using the eigenfunction of $A$ with eigenvalue $\lambda$. We will define the sphere matrix $\mathbf{S}^{(r)}$ to compute the following matrix-vector product:

$$\mathbf{S}^{(r)} f(x) = \sum_{y \in \mathbf{S}^{(r)}(x)} f(y)$$

One can then prove that if $\varphi_i$ is an eigenfunction of matrix $A$ with eigenvalue $\lambda_i$ then $\varphi_i$ is also and eigenfunction of matrix $\mathbf{S}^{(r)}$ with eigenvalue $\gamma_i^{(r)}$. The calculation of $\gamma_i^{(r)}$ corresponds to the well-known Krawtchouk polynomials [10] which have the following closed form solution:

$$\gamma_i^{(r)} = \sum_{j=0}^{|i|} \binom{|i|}{j} \binom{n-|i|}{r-j} (-1)^j$$

This equation allows us to compute neighborhood averages over arbitrary Hamming spheres in the MAX-kSAT search graph.

$$\mu_c(\mathbf{S}^{(r)}(x)) = \binom{n}{r} - 1 \sum_j \gamma_i^{(r)} \omega_j \psi_j(x)$$

This result generalizes over arbitrary Hamming balls by summing over nested spheres [17].

#### 2.3.3 Directed Plateau Search

On many combinatorial problems, hill-climbing algorithms must contend with plateaus: vast regions of the search space containing states with only equal or disimproving neighbors. On plateaus, local search algorithms can no longer utilize any gradient information so they resort to selecting equal valued neighbors at random until an improving neighbor state is encountered (or they can accept disimproving moves). Local search algorithms for SAT such as Walksat do exactly this. These algorithms have been surprisingly successful even though plateaus can grow exponentially as the optimal is approached; yet they do sometimes get stuck.

We have created Directed Plateau Search (DPS) [15] to address the plateau problem by extracting principled information from the search space to make more informed plateau moves. DPS computes the average value of the points in a Hamming ball around a given point in the search space for arbitrary MAX-kSAT problems. Assume that we have a point $x$ in the search space, and that points $x_1$ and $x_2$ are neighbors of $x$. Also assume that $f(x_1) = f(x_2)$ so that all the points are on a plateau. This idea is illustrated in Figure 2.

The current implementation of DPS computes the average value of the points in a Hamming ball of radius 5. This value is then treated as a "tie-breaker" in the absence of a gradient to follow; the idea is that a state with a comparatively better average value in this generalized

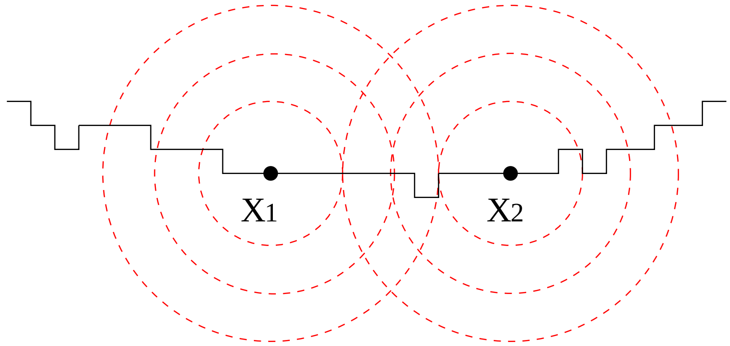


Figure 2: Assume we have two solutions, $x_1$ and $x_2$ that are on a plateau and have the same evaluation. We can compute the statistical moments around localized Hamming balls. The Hamming ball at distance 1 is the usual bit-flip neighborhood. While $x_1$ and $x_2$ have the same evaluation, the statistic moments for different Hamming balls around the points are likely to differ.

neighborhood is more likely to contain nearby plateau escapes. Thus, DPS is used to augment existing hill-climbing search algorithms by using information about the localized Hamming ball, rather than resorting to a random walk.

Figure 3 shows data from two problem instances, one where GWSAT+DPS is best and another where GWSAT alone was best. This is typical of the behaviors we have seen on several benchmark and randomly generated problems. Preliminary evidence suggests that when the information found in the averages of the Hamming balls allows for strong discrimination, DPS is very helpful. This is particularly true during the earlier stages of search. In figure 4 we terminated search when a solution with an error of 2 was found (all but 2 clauses are satisfied); in these tests, GWSAT+DPS nearly always is the best algorithm compared to all of the SAT-Solvers we tested.

#### 2.3.4 Autocorrelation

Properties beyond the statistical moments may also prove beneficial for guiding search. Measurements such as autocorrelation have been suggested as metrics that can predict problem difficulty. The autocorrelation is usually calculated by using a random walk to estimate the following equation

$$r(s) = \frac{\langle f, \mathbf{T}^s f\rangle - \langle \mathbf{1}, f\rangle^2}{\langle f, f\rangle - \langle \mathbf{1}, f\rangle^2}$$

However, we have the following identities:

$$\langle f, f\rangle = \sum_i w_i^2 \qquad \langle f, \mathbf{T}^s f\rangle = \sum_i \lambda_i^s w_i^2 \qquad \langle \mathbf{1}, f\rangle = w_0$$

This allows us to compute the *exact* autocorrelation for any MAX-kSAT problem [16].

$$r(s) = \frac{\sum_{i\neq 0} \lambda_i^s w_i^2}{\sum_{j\neq 0} w_j^2} \quad \text{where} \quad \lambda_i = \left(1 - \frac{2\langle i, i\rangle}{n}\right)$$

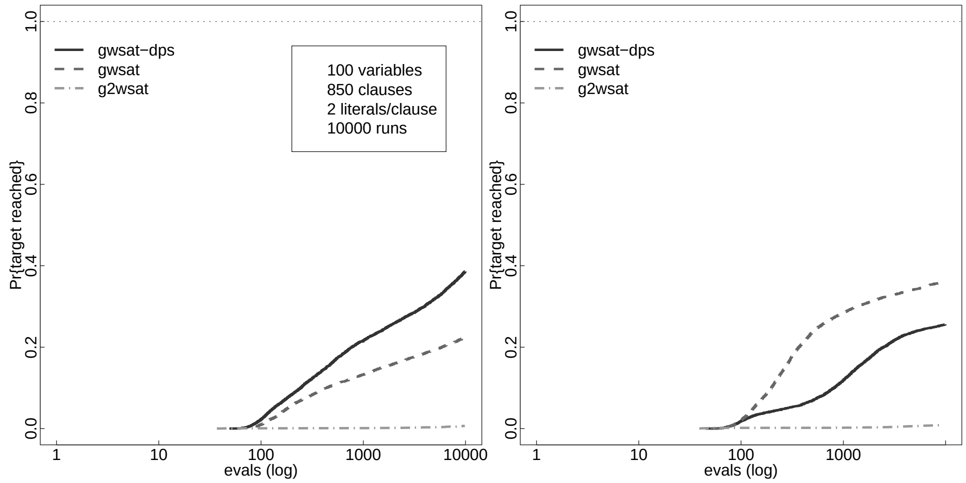


Figure 3: This graph presents different results for two MAXSAT problems. GWSAT+DPS or GWSAT was superior to all of the other SAT-Solvers we tested across many problems. About half of the time GWSAT+DPS was best, and about half the time GWSAT was best. The graphs measure the frequency and speed with which a global optimum was found across 10,000 attempts. The results shown here are only for a 100 variable problem which can also be solved with DPLL, but we have run experiments on a variety of benchmark and random problems with similar results.

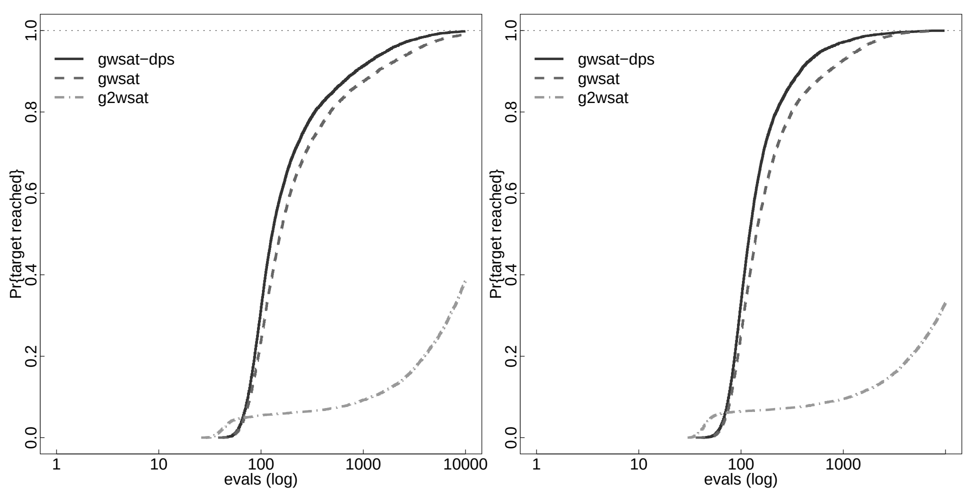


Figure 4: In this case, we measure the amount of time it took to find a solution with all but 2 clauses satisfied. At this threshold, GWSAT+DPS was almost always superior to all other SAT-Solvers we tested. The graphs measure the frequency and speed with which a solution with the target error optimum was found across 10,000 attempts. This shows that GWSAT+DPS is superior in the early stages of search; sometimes it goes on to quickly find a solution, but sometimes it gets stuck. This suggest that we might need to make DPS more focused and localized by using a smaller radius as it finds better solutions.

# 3 Executive Summary

## 3.1 Personnel

During the grant period, the following personnel were supported at the indicated level:

| | | | |
|---|---|---|---|
| PIs: | | | |
| | Adele Howe | 4.5 | months |
| | L. Darrell Whitley | 0 | months |
| Research Assistants: | | | |
| | Doug Hains | 10.75 | full-time months (13.5 half-time, 4 full-time) |
| | Andrew Sutton | 15.5 | full-time months (18 half-time, 6.5 full-time) |

## 3.2 Publications

### Journals

- A. Sutton, L.D. Whitley and A. Howe. Computing the moments of k-bounded pseudo-Boolean functions over Hamming spheres of arbitrary radius in polynomial time, to appear in *Theoretical Computer Science* journal.
- F. Chicano, D. Whitley, E. Alba, "A Methodology to Find the Elementary Landscape Decomposition of Combinatorial Optimization Problems". To appear in *Evolutionary Computation.*
- F. Chicano, D. Whitley, E. Alba, F Luna, "Elementary Landscape Decomposition of the Frequency Assignment Problem". To appear in *Theoretical Computer Science.*
- Rinku Dewri, Indrakshi Ray, Indrajit Ray, D. Whitley "k-Anonymization in the Presence of Publisher Preferences". to appear in *Transactions on Knowledge and Data Engineering.*
- Doug Hains, L. Darrell Whitley, Adele E. Howe. Revisiting the Big Valley Search Space Structure in the TSP, in *Journal of Operations Research Society*, Vol. 62, pp. 305312, September 2010.
- M. Roberts and A.E. Howe. 2009. Learning from Planner Performance, *Artificial Intelligence*, Vol. 173, Issues 5-6, pp. 536-561, April.

### Conferences and Workshops

- R. Dewri and L.D. Whitley, "A Multi-objective approach to data sharing with privacy constraints and preference based objects." In *Proceedings of GECCO 2009*, Montreal, CA, July 2009.
- A.M. Sutton, L.D. Whitley and A.E. Howe, "A polynomial time computation of the exact correlation structure of k-satisfiability landscapes", In *Proceedings of GECCO 2009*, Montreal, CA, July 2009.
- L.D. Whitley, D. Hains and A.E. Howe, "Tunneling between Optima: Partition Crossover for the TSP", In *Proceedings of GECCO 2009*, Montreal, CA, July 2009.
- L.D. Whitley and A. Sutton, "Partial Neighborhoods of Elementary Landscapes." In *Proceedings of GECCO 2009*, Montreal, CA, July 2009.

- A. M. Sutton, A. E. Howe, and L. D. Whitley. "Directed Plateau Search for MAX-k-SAT". In *Proceedings of the Third Annual Symposium on Combinatorial Search*, Atlanta, GA, July 2010.

- A.M. Sutton, A.E. Howe and L.D. Whitley, "A theoretical analysis of the k-satisfiability search space", in *Proceedings of Stochastic Local Search 2009 Workshop* and *Lecture Notes in Computer Science*, Vol. 5752, pp.46-60, Brussels, Belgium, September 2009.

- A.M. Sutton, A.E. Howe and L.D. Whitley, "Estimating Bounds on Expected Plateau Size in MAXSAT Problems" in *Proceedings of Stochastic Local Search 2009 Workshop* and *Lecture Notes in Computer Science*, Vol. 5752, pp.31-45, Brussels, Belgium, September 2009.

- D. Whitley, F. Chicano, E. Alba, F. Luna, "Elementary Landscapes of Frequency Assignment Problems." *Proceedings of GECCO-2010.* ACM Press.

- L.D. Whitley, D. Hains and A. Howe, A Hybrid Genetic Algorithm for the Traveling Salesman Problem using Generalized Partition Crossover, in *Proceedings of Parallel Problem Solving from Nature XI*, September 2010.

- R. Dewri, I. Ray, I. Ray and D. Whitley. "Historical k-Anonymous Anonymity Sets in a Continuous LPS", In *Proceedings of International Conference on Security and Privacy in Communication Networks (SecureComm)*, Singapore, 2010.

- R. Dewri, I. Ray, I. Ray and D. Whitley. "On the Identification of Property Based Generalization in Microdata Anonymization," *24th IFIP WG 11.3 Working Conference on Data and Applications Security (DBSec)*, pp. 81-96, Rome, Italy.

- A. Sutton, L.D. Whitley and A.E. Howe, Approximating the Distribution of Fitness over Hamming Regions, in *Proceedings of Foundations of Genetic Algorithms (FOGA-2011)*, January 2011.

**Other Publications**

- Doug Hains. "Generalized Partition Crossover for the Traveling Salesman Problem", Master's thesis, Colorado State University, Colorado, December 2010.

- Doug Hains, Adele Howe and Darrell Whitley. "Smoothing Funnels in the TSP with Genetic Algorithms". in *Proceedings of Colorado Celebration of Women in Computing*, November 2010.

Note: Andrew Sutton, one of the graduate students funded from the grant is scheduled to defend his Ph.D. thesis, predominantly research conducted using this funding, on March 21, 2011.

## 3.3 Interactions/Transitions

### 3.3.1 Presentations at Meetings

**A. Sutton** Oral presentation of *Approximating the Distribution of Fitness over Hamming Regions* at Foundations of Genetic Algorithms XI, Schwarzenberg, Austria, January 2011; *Local Statistics of Bounded Pseudo-Boolean Functions.* at Dagstuhl Seminar on the Theory of Evolutionary Algorithms. Schloss Dagstuhl, Germany, September 2010; *Directed*

*Plateau Search for MAX-k-SAT.* at Third Annual Symposium on Combinatorial Search, Atlanta, GA, July 2010; *A Theoretical Analysis of the k-Satisfiability Search Space.* at Second Workshop on Engineering Stochastic Local Search Algorithms. Brussels, Belgium, 4 September 2009; *Estimating Bounds on Expected Plateau Size in MAXSAT Problems.* at Second Workshop on Engineering Stochastic Local Search Algorithms. Brussels, Belgium, 3 September 2009; *Elementary Landscapes: On the semi-decomposibility of select* NP*-hard optimization problems* (with L. D. Whitley). at Evolutionary Computation Day in Birmingham, Centre of Excellence for Research in Computational Intelligence and Applications, Birmingham, UK, 10 August 2009; *A Polynomial Time Computation of the Exact Correlation Structure of k-Satisfiability Landscapes.* at Genetic and Evolutionary Computation Conference. Montreal, QC, 10 July 2009; *On the semi-decomposibility of select* NP*-hard optimization problems* (with L. D. Whitley) at Colorado State University Mathematics Colloquium. Fort Collins, CO, 13 April 2009. Invited talk entitled *Modeling Combinatorial Search Spaces.* at Computer Science Research Institute, Sandia National Laboratory, Albuquerque, NM, June 2010.

**A. Sutton and L.D. Whitley** Presented tutorial entitled *Elementary Landscape Analysis for TSP, Graph Coloring, Graph Partitioning, and MAXSAT* (with L. D. Whitley) at Genetic and Evolutionary Computation Conference. Montreal, QC, 9 July 2009.

**A. Howe and L.D. Whitley** "Extensions to the Theory of Elementary Landscapes" talk at annual AFOSR PI meeting, April 21, 2009 in Arlington, VA; "Exploiting Elementary Landscapes" talk at annual AFOSR PI meeting, April 19, 2010 in Arlington, VA.

**A. Howe** invited talk entitled "Artificial Intelligence: Fact versus Fiction" for College of Natural Sciences, Colorado State University, November 10, 2010. Oral presentation of What Makes Planners Predictable?, at ICAPS 2008 in Sydney, Australia, September 2008.

**L.D. Whitley** keynote lecture at Parallel Problem Solving from Nature Conference 2010 in Krakow, Poland; keynote lecture at The 9th International Conference on Artificial Evolution in Strasbourg in 2009; keynote lecture at 2009 GECCO Workshop on Self Guided Metaheuristics, Montreal.

### 3.3.2 Consultative Functions at Laboratories and Agencies

A. Howe is currently (2010-2011) a member of a Defense Science Board Task Force investigating Autonomy.

### 3.3.3 Transitions

**FirstRF Corp** worked with team on SBIR Phase 1 grant for developing the next generation AFSCN system.

## 3.4 Honors/Awards/Significant Service

**D. Whitley, D. Hains, A. Howe** Best paper award for Tunneling between Optima: Partition Crossover for the TSP at GECCO 2009.

**A. Howe** Member Defense Science Board Task Force on Autonomy (2010-2011); Designated Professor Laureate for College of Natural Sciences, Colorado State University 2010-2012;

Member of Executive Council for AAAI professional society 2010-2013; Secretary of ICAPS Executive Council 2009-2015; Program Co-Chair for International Conference on Automated Planning and Scheduling (ICAPS 2009); Associate Editor for *Journal of Artificial Intelligence Research* (JAIR) 2007-2009; Member Advisory Board for JAIR 2010-2012.

**L.D. Whitley** Chair of the Governing Board for ACM Sigevo. Member of the ACM Sig Governing Board. Co-Chair of a workshop for new CS department heads at the Computing Research Association "Snowbird" conference. Associate Editor for *Theoretical Computer Science, Evolutionary Computation* and *Artificial Intelligence*. Editorial Board, *Journal of Heuristics*. Advisory Board, *Springer Natural Computation Series*. Organizer of Dagstuhl Workshop on "Theory of Evolutionary Algorithms" in 2009. Honorary Program Co-Chair, Genetic and Evolutionary Computation Summit 2009.

### 3.5 Web Site

Our project web site is available at `http://www.cs.colostate.edu/sched/`. From that site, you can access publications and data from the project.

## References


[1] D. Applegate, W. Cook, and A. Rohe. Chained Lin-Kernighan for large traveling salesman problems. *INFORMS Journal on Computing*, 15(1):82–92, 2003.

[2] J. W. Barnes, B. Dimova, and S. P. Dokov. The theory of elementary landscapes. *Applied Mathematics Letters*, 16(3):337–343, April 2003.

[3] F. Chicano, D. Whitley, E. Alba, and F. Luna. Elementary landscape decomposition of the frequency assignment problem. *Theoretical Computer Science*, In Press.

[4] W. Cook and P. Seymour. Touring Merging via Branch Decomposition. *INFORMS Journal on Computing*, 15(3):233–248, 2003.

[5] GA Croes. A method for solving traveling-salesman problems. *Operations Research*, pages 791–812, 1958.

[6] Robert B. Heckendorn. Embedded landscapes. *Evolutionary Computation*, 10(4):345–369, 2002.

[7] K. Helsgaun. An effective implementation of the lin-kernighan traveling salesman heuristic. *European Journal of Operational Research*, 126(1):106–130, 2000.

[8] K. Helsgaun. General k-opt submoves for the lin-kernighan tsp heuristic. *Mathematical Programming Computation*, 1(2-3), 2009.

[9] D. S. Johnson and L. A. McGeoch. The traveling salesman problem: A case study in local optimization. In E. H. L. Aarts and J.K. Lenstra, editors, *Local Search in Combinatorial Optimization*, pages 215–310. John Wiley and Sons Ltd, 1997.

[10] V.I. Levenshtein. Krawtchouk polynomials and universal bounds for codes and designs in hamming space. *IEEE Transactions on Information Theory*, 2002.

[11] S. Lin and B. Kernighan. An effective heuristic algorithm for the traveling salesman problem. *Operations Research*, 21:498–516, 1973.

[12] Soraya Rana, Robert B. Heckendorn, and L. Darrell Whitley. A tractable Walsh analysis of SAT and its implications for genetic algorithms. In *Proceedings of the Fifteenth National Conference on Artificial Intelligence (AAAI'98)*, pages 392–397, 1998.

[13] Peter F. Stadler. Landscapes and their correlation functions. *Journal of Mathematical Chemistry*, 20:1–45, 1996.

[14] Andrew M. Sutton, Adele E. Howe, and L. Darrell Whitley. A theoretical analysis of the k-satisfiability search space. In *SLS*, pages 46–60, 2009.

[15] Andrew M. Sutton, Adele E. Howe, and L. Darrell Whitley. Directed plateau search for max-k-sat. In *Proceedings of the Third Annual Symposium on Combinatorial Search*, Atlanta, GA, 2010.

[16] Andrew M. Sutton, L. Darrell Whitley, and Adele E. Howe. A polynomial time computation of the exact correlation structure of k-satisfiability landscapes. In *Proceedings of the Genetic and Evolutionary Computation Conference (GECCO-09)*, 2009.

[17] Andrew M. Sutton, L. Darrell Whitley, and Adele E. Howe. Computing the moments of $k$-bounded pseudo-Boolean functions over Hamming spheres of arbitrary radius in polynomial time. *Theoretical Computer Science*, 2011. In press.

[18] D. Whitley, F. Chicano, E. Alba, and F. Luna. Elementary landscapes of frequency assignment problems. In *Proceedings of the Genetic and Evolutionary Computation Conference (GECCO-2010)*, 2010.

[19] D. Whitley, D. Hains, and A. Howe. A hybrid genetic algorithm for the traveling salesman problem using generalized partition crossover. In *PPSN-2010*. Springer, 2009.

[20] D. Whitley and A. Sutton. Partial neighborhoods of elementary landscapes. In *GECCO-2009*, 2009.

[21] L. Darrell Whitley, Andrew M. Sutton, and Adele E. Howe. Understanding elementary landscapes. In *Proceedings of the Genetic and Evolutionary Computation Conference*, Atlanta, GA, July 2008.